FOP Design: Input Parsing

$Revision: 426576 $

Table of contents

1. Introduction

Parsing is the process of reading the XSL-FO input and making the information in it available to FOP.

2. SAX for Input

The two standard ways of dealing with XML input are SAX and DOM. SAX basically creates events as it parses an XML document in a serial fashion; a program using SAX (and not storing anything internally) will only see a small window of the document at any point in time, and can never look forward in the document. DOM creates and stores a tree representation of the document, allowing a view of the entire document as an integrated whole. One issue that may seem counter-intuitive to some new FOP developers, and which has from time to time been contentious, is that FOP uses SAX for input. (DOM can be used as input as well, but it is converted into SAX events before entering FOP, effectively negating its advantages).

Since FOP essentially needs a tree representation of the FO input, at first glance it seems to make sense to use DOM. Instead, FOP takes SAX events and builds its own tree-like structure. Why?

- DOM has a relatively large memory footprint. FOP's FO Tree is a lighter-weight structure.
- DOM contains an entire document. FOP is able to process individual fo:page-sequence objects discretely, without the need to have the entire document in memory. For documents that have only one fo:page-sequence object, FOP's approach is no advantage, but in other cases it is a huge advantage. A 500-page book that is broken into 100 5-page chapters, each in its own fo:page-sequence, essentially needs only 1% of the document memory that would be required if using DOM as input.

See the Input Section of the User Embedding Document for a discussion of input usage patterns and some implementation details.

FOP's FO Tree Mechanism is responsible for catching the SAX events and processing them.

3. Validation

If the input XML is not well-formed, that will be reported.

There is no DTD for XSL-FO, so no formal validation is possible at the parser level.

The SAX handler will report an error for unrecognized namespaces.

4. Namespaces

To allow for extensions to the XSL-FO language, FOP provides a mechanism for handling foreign namespaces.

See User Extensions for a discussion of standard extensions shipped with FOP, and their related namespaces.

See Developer Extensions for a discussion of the mechanisms in place to allow developers to add their own extensions, including how to tell FOP about the foreign namespace.

5. Status

5.1. To Do

5.2. Work In Progress

5.3. Completed

- better handling of unknown xml and xml from an unknown namespace
- Changed extensions to allow for external xml
- Can have a default element mapping for extensions